PTO/SB/33 (07-05)
Approved for use through xx/xx/200x. OMB 0651-00xx
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PRE-APPEAL BRIEF REQUEST FOR REVIEW		Docket Number: 10559-913001 / P18139
I hereby certify under 37 CFR §1.8(a) that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop AF, Commissioner for Patents, Box 1450, Alexandria, VA 22313-1450. Date of Deposit Signature Typed or Printed Name of Person Signing Certificate	Application Number 10/797,886	Filed March 9, 2004
	First Named Inventor Bratin Saha	
	Art Unit 2183	Examiner Brian P. Johnson

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a Notice of Appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)

☒ attorney or agent of record 47,671
(Reg. No.)

☐ attorney or agent acting under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34

Signature

William E. Hunter
Typed or printed name

(858) 678-5070
Telephone number

June 6, 2008
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below'.

☒ Total of one form (plus five attached sheets) are submitted.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Bratin Saha	Art Unit	: 2183
Serial No.	: 10/797,886	Examiner	: Brian P. Johnson
Filed	: March 9, 2004	Conf. No.	: 5086
Title	: SYNCHRONIZATION OF PARALLEL PROCESSES		

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Pursuant to the Pre-Appeal Brief Conference Program, a request for a review of identified matters on appeal is hereby submitted in view of clear legal or factual deficiencies in the rejections. All rights to address additional matters in the full appeal brief are hereby reserved.

Claims 11-34 are pending, with claims 11, 16, 21, 25 and 33 being independent. Claims 11-13, 15-18, 20, 21, 23-27, 29, 30 and 34 stand rejected under 35 U.S.C. §102(b) as allegedly being anticipated by the Lock Reservation paper (hereinafter "Koseki"). Claim 33 stands rejected under 35 U.S.C. §103(a) as allegedly being anticipated over Koseki, in view of the Transactional Memory paper (hereinafter "Moss"). Claims 14, 19, 22, 28 and 32 stand rejected under 35 U.S.C. §103(a) as allegedly being anticipated over Koseki, in view of "Common Art".

Independent claim 11 recites (emphasis added), "generating parallel processes in a data processing machine; effecting synchronization between the parallel processes using processor speculation in the data processing machine to speculatively execute one or more instructions that read-modify-write a lock variable associated with a critical section and end speculation before performing the critical section; and providing output resulting from the synchronized parallel processes." The Office cites to the Abstract of Koseki as allegedly teaching this subject matter, but nothing in Koseki's Abstract (or any other portion of Koseki) teaches or suggests using processor speculation to speculatively read-modify-write a lock variable. The Office's position that anytime a processor performs operations that may later be discarded, this constitutes "speculative" processing as claimed, cannot be maintained since it contradicts the plain meaning of the claim language, in light of the knowledge of those of ordinary skill in the art and the present Specification.[1]

[1] *See e.g.,* Specification at page 4, ¶ 13.

Claim 11 expressly recites (emphasis added), "using processor speculation in the data processing machine to speculatively execute one or more instructions that read-modify-write a lock variable". This clearly refers to the use of the speculative execution capabilities of modern processors. Koseki does not in any way teach or suggest such speculative execution of one or more instructions that read-modify-write a lock variable. In response, the Office mistakenly asserts that the above citation to the present Specification is a citation to Koseki, and asserts that Koseki's use of the phrase "speculative execution" is consistent with the Office's interpretation.[2] This is incorrect. Koseki **does** describe speculative execution of machine instructions in a manner consistent with Applicant's interpretation: "Kung and Robinson [29] proposed an optimistic concurrency control for database systems, which speculatively executes critical regions without acquiring locks and commits the changes if there is no contention."[3] Koseki **does not** describe speculatively executing one or more instructions that read-modify-write a lock variable associated with a critical section. Rather, Koseki describes a lock reservation scheme that exploits the observation that Java locks show thread locality[4] by doing the following. The lock is first anonymously reserved for the initial locker[5], and when the initial locker attempts to acquire the anonymously reserved lock, the initial locker attempts to make this reservation specific by adding its thread identifier (TID) to the reserve word using an atomic compare_and_swap operation.[6] As long as each subsequent attempt to acquire the lock is made by the same thread that made the reservation specific (i.e., there is thread locality as shown in Figure 1 of Koseki), that thread can acquire the lock without using an atomic operation.[7]

Thus, that initial locker thread can continue to acquire the lock, execute its critical section(s) and release the lock, multiple times, without having to use an atomic operation for each of the multiple lock acquisitions. But, as soon as a new thread comes along with an attempt to acquire the lock, the unreserve() function is called to cancel the reservation[8], "requiring the owner thread to be suspended."[9] This suspending process can happen when the first thread is in

---

[2] *See* 2-6-2008 Office Action at p. 11.
[3] *See* Koseki at p. 139.
[4] *See* Koseki at page 132, Sec. 3, 1st ¶.
[5] *See* Koseki at page 132, Sec. 3.1, 3rd ¶, and Figure 2(c).
[6] *See* Koseki at page 133, Sec. 3.2, 3rd ¶, and page 134 at lines 22 & 31-34 of the algorithm code.
[7] *See* Koseki at page 133, Sec. 3.2, 1st ¶, and page 134 at lines 21-29 of the algorithm code.
[8] *See* Koseki at page 133, Sec. 3.2, 3rd ¶, and page 134 at lines 23 & 37 of the algorithm code.
[9] *See* Koseki at page 132, Sec. 3.1, 5th ¶; page 133, Sec. 3.2, 5th & 6th ¶s, and page 134 at line 74 of the code.

one of the unsafe regions, which requires special handling.[10] This special handling is accomplished using software[11], **not** processor speculation, as claimed.

The lock reservation scheme described in Koseki cannot be considered to be **speculative execution** of the one or more instructions that read-modify-write a lock variable because Koseki clearly notes what speculative execution is in connection with his reference to Kung and Robinson, and Koseki never describes his lock reservation scheme as using **processor speculation** or **speculative execution**. The Office's refusal to recognize the well established meaning in the art of using **processor speculation** in a **data processing machine** to **speculative execute instructions** is completely without justification, and the Office's interpretation of the scope of the speculative execution claim language is improper. The Office refuses to recognize that the current claim language necessarily imposes a requirement that the speculative execution capabilities of modern processors be used[12], even though this is the plain meaning of the claim language in light of the Specification, the cited reference itself and the knowledge of those of ordinary skill in the art. Thus, for all of the above reasons, the rejection of claim 11 suffers from clear legal or factual deficiencies and should be withdrawn.

Independent claim 16 recites (emphasis added), "generating parallel processes in a data processing machine; effecting synchronization between the parallel processes using processor speculation in the data processing machine to speculatively execute one or more instructions that read-modify-write a lock variable associated with a critical section and end speculation before performing the critical section; and providing output resulting from the synchronized parallel processes." Thus, based on reasons similar to those presented above, the rejection of claim 16 suffers from clear legal or factual deficiencies and should be withdrawn.

In addition, claims 13 and 18 each recite, "wherein said effecting synchronization comprises translating at least one high-level software instruction into at least one machine instruction that controls speculative execution in a processor." The cited portion of Koseki (page 130, footnote 1) says nothing about a machine instruction that controls speculative execution in a processor, as pointed out in the Response filed 11-20-2007. Thus, the rejection of claims 13 and 18 suffers from additional, clear legal or factual deficiencies and should be withdrawn.

---

[10] *See* Koseki at page 133, Sec. 3.2, 6th and 7th ¶s.
[11] *See* Koseki at page 135, Sec. 3.4, subsection "Unsafe Regions".
[12] *See* 2-6-2008 Office Action at p. 12.

Independent claim 21 recites (emphasis added), "speculatively executing machine instructions, including a memory access instruction, in a processing system to effect synchronization between parallel processes, wherein the speculatively executing comprises performing a speculative read-modify-write to a lock variable associated with a critical section; retiring the speculatively executed machine instructions to end speculation before performing the critical section; and maintaining cache coherence in the processing system during said executing and said retiring to identify a mis-speculation to effect the synchronization between the parallel processes." The rejection of claim 21 should be withdrawn for at least reasons similar to those presented above with respect to claim 11. Moreover, nothing in Koseki describes maintaining cache coherence in a processing system to identify a mis-speculation to effect synchronization between the parallel processes, as pointed out previously.[13] Thus, the rejection of claim 21 suffers from an additional, clear legal or factual deficiency and should be withdrawn.

Independent claim 25 recites (emphasis added), "a processor having a processor architecture that provides speculative execution of machine instructions and exposes said speculative execution to program control through at least one machine instruction; and a memory coupled with the processor, the memory embodying information indicative of instructions, including the at least one machine instruction, that result in synchronization between parallel processes when performed by the processor with detection of mis-speculation; wherein performance of the instructions by the processor comprises performing a speculative read-modify-write to a lock variable associated with a critical section and ending speculation before performing the critical section." The Office has failed to provide any evidence that Koseki describes a processor architecture that provides speculative execution of machine instructions and exposes said speculative execution to program control through at least one machine instruction. Thus, based on this, and on reasons similar to those presented above, the rejection of claim 25 suffers from clear legal or factual deficiencies and should be withdrawn.

Independent claim 33 recites (emphasis added), "processing means for speculatively executing machine instructions in response to a speculative execution instruction, including means for detecting a mis-speculation; means for treating multiple speculative instructions as a group for purposes of retirement such that the multiple speculative instructions are flushed from

[13] *See* 11-20-2007 Response at page 6.

the processing means together and execution proceeds from an address in response to a detected mis-speculation to effect synchronization between parallel processes; wherein performance of the instructions by the processing means comprises performing a speculative read-modify-write to a lock variable associated with a critical section and ending speculation before performing the critical section." Moss fails to cure the noted deficiencies of Koseki. Moss teaches "lock-free" synchronization, which is mutually exclusive of using processor speculation in a data processing machine to speculatively read-modify-write a lock variable associated with a critical section, as claimed. For synchronization to be **lock-free**, there must be **no lock**. Thus, the proposed combination of references is illogical and does not teach or suggest the claimed subject matter.

Moreover, to the extent that Koseki's lock reservation technique can be considered to teach a form of "speculation", which is not conceded, the "speculation" and detection of "mis-speculation" in Koseki is all done in software[14]. Nothing in Koseki suggests the use of hardware speculation, as the present means-for language necessarily entails. Thus, the rejection of claim 33 also suffers from clear legal or factual deficiencies and should be withdrawn.

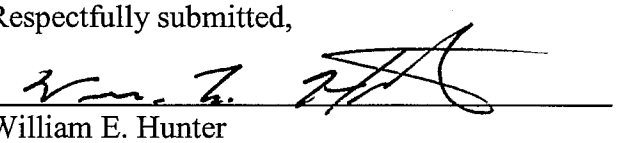
Dependent claim 34 recites, "wherein said means for detecting a mis-speculation comprises means for maintaining cache coherence in the processing means." The Office appears to refer to Koseki's "monitor cache" in rejecting this claim. However, the monitor cache in Koseki is a software data structure that is clearly distinct from the hardware cache referred to by claim 34. Thus, the rejection of claim 34 also suffers from clear legal or factual deficiencies and should be withdrawn.

In view of the above, all of the claims should be in condition for allowance. A formal notice of allowance is thus respectfully requested.

Please apply any necessary charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: June 6, 2008



William E. Hunter
Reg. No. 47,671

Fish & Richardson P.C.
Telephone: (858) 678-5070
Facsimile: (858) 678-5099
PTO Customer No. **20985**

10826546.doc

[14] *See* Koseki at page 135, Sec. 3.4, subsection "Unsafe Regions".